Experiment 02 - Exploratory Data Analysis and Visualization

| Roll No. | 65 |
| --- | --- |
| Name | Shreyansh Singh |
| Class | D15 |
| Subject | Business Intelligence Lab |
| LO Mapped | LO1:  Identify sources of Data for mining and perform data exploration |
|  |  |

**Aim**: To perform exploratory data analysis and visualization on the dataset using python.

**Introduction**:

EDA stands for Exploratory Data Analysis. It is a process of examining and analyzing data sets to summarize their main characteristics, often with visual methods. EDA is typically used to analyze the data before applying any specific modeling techniques or algorithms to the data. The goal of EDA is to gain insights and understanding of the data, identify patterns, relationships, and trends, and detect potential anomalies or outliers. EDA can involve various statistical and visual techniques such as histograms, box plots, scatter plots, correlation matrices, and more. The results of EDA can guide the selection of appropriate data preprocessing and modeling techniques, and can help in making decisions and predictions based on the data.

**Descriptive analysis - Central tendency**

Central tendency is a statistical measure that represents the middle or central value of a data set. There are three commonly used measures of central tendency: mean, median, and mode.

Mean is the arithmetic average of a data set and is calculated by adding up all the values and dividing by the number of observations. Mean is affected by outliers and extreme values and can be skewed if the distribution is not normal.

Median is the middle value of a data set and is calculated by arranging the observations in ascending or descending order and selecting the value in the middle. Median is not affected by outliers or extreme values and is a more robust measure than mean.

Mode is the most common value in a data set and is calculated by finding the value that occurs most frequently. Mode is useful for identifying the most typical or representative value in a data set.

To demonstrate how central tendency can be identified, we can use the following code to calculate the mean, median, and mode for a given dataset:

import numpy as np

# generate sample data

data = np.random.normal(loc=10, scale=2, size=1000)

# calculate mean

mean = np.mean(data)

# calculate median

median = np.median(data)

# calculate mode

mode = np.mode(data)

print("Mean:", mean)

print("Median:", median)

print("Mode:", mode)

In the above code, we generate a sample dataset using numpy.random.normal function. We then calculate the mean, median, and mode using the numpy.mean, numpy.median, and scipy.stats.mode functions, respectively. Finally, we print the values of the calculated measures of central tendency.

With respect to our dataset, we can identify the central tendency of the relevant features to gain insights into the distribution of data. For example, we can calculate the mean, median, and mode of the prices of houses in a particular area to understand the average price, the price at which half of the houses are sold, and the most frequent price range. These measures can be useful for making decisions related to buying or selling houses in that area.

**Descriptive analysis - Dispersion**

Data dispersion refers to the spread or variability of a dataset. It can be measured using various statistical measures, including range, variance, standard deviation, and interquartile range. Here are the steps to identify data dispersion in a dataset:

Range:

Range is the simplest measure of dispersion and is calculated by subtracting the minimum value from the maximum value in the dataset. The range provides an idea about the spread of the data, but it is sensitive to outliers.

Variance and standard deviation:

Variance is the average of the squared differences from the mean, while standard deviation is the square root of variance. These measures provide information about how far the data points are from the mean and how much they vary from each other.

Interquartile range (IQR):

IQR is the range of the middle 50% of the data points and is calculated by subtracting the first quartile (Q1) from the third quartile (Q3). IQR provides a robust measure of dispersion that is less sensitive to outliers.

**Correlation**

Correlation analysis is a statistical method used to measure the strength and direction of the relationship between two variables. In data analysis, we often use correlation analysis to identify correlations between different attributes in a dataset. Here are the steps to identify correlations between different attributes:

Load the data:

Load the dataset into a software or programming language of your choice.

Calculate correlation matrix:

Calculate the correlation matrix of the dataset, which provides the correlation coefficients between each pair of attributes. There are several correlation coefficients that can be used, including Pearson's correlation coefficient, Spearman's correlation coefficient, and Kendall's Tau correlation coefficient.

Visualize the correlation matrix:

Visualize the correlation matrix using a heatmap or scatter plot matrix. This will help to identify strong and weak correlations between different attributes.

Analyze the correlation matrix:

Analyze the correlation matrix to identify the attributes that are highly correlated. A correlation coefficient of 1 indicates a perfect positive correlation, while a correlation coefficient of -1 indicates a perfect negative correlation. A correlation coefficient of 0 indicates no correlation.

Here's an example Python code snippet using Pandas and Seaborn to calculate the correlation matrix and visualize it as a heatmap:

import pandas as pd

import seaborn as sns

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Calculate the correlation matrix

corr\_matrix = data.corr()

# Visualize the correlation matrix as a heatmap

sns.heatmap(corr\_matrix, annot=True)

Inference with respect to the dataset can be made by analyzing the correlation matrix and the visualization. A high correlation coefficient between two attributes indicates that they are highly correlated, while a low correlation coefficient indicates a weak correlation.

For example, if we are analyzing a dataset of housing prices, we may find that the size of the house and the number of bedrooms are highly correlated, indicating that larger houses tend to have more bedrooms. We may also find that the age of the house and the price are negatively correlated, indicating that older houses tend to be less expensive.

In summary, identifying correlations between different attributes is an important step in data analysis, as it helps to understand the relationships between different variables in the dataset. By using appropriate statistical techniques and visualization tools, we can gain insight into the dataset and make informed decisions about data analysis techniques.

**Data Visualization**

Visualization is an important step in data analysis that helps to gain insights into the data and understand the relationships between different variables. Here are five visualization techniques that can be used in data analysis:

Histogram:

A histogram is a graphical representation of the distribution of a dataset. It shows how often each value in the dataset falls within a certain range or bin. Histograms are useful for identifying patterns in the data, such as the presence of outliers or the skewness of the data.

Example Python code using Matplotlib to create a histogram:

import matplotlib.pyplot as plt

import pandas as pd

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Create a histogram of the values in the 'age' column

plt.hist(data['age'])

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.show()

Inference: The histogram can help us understand the age distribution in the dataset and identify any patterns, such as a bimodal distribution or outliers.

Scatter plot:

A scatter plot is a graphical representation of the relationship between two variables. It shows how the values of one variable are related to the values of another variable. Scatter plots are useful for identifying patterns and correlations between variables.

Example Python code using Matplotlib to create a scatter plot:

import matplotlib.pyplot as plt

import pandas as pd

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Create a scatter plot of the 'age' and 'income' columns

plt.scatter(data['age'], data['income'])

plt.xlabel('Age')

plt.ylabel('Income')

plt.show()

Inference: The scatter plot can help us understand the relationship between age and income in the dataset and identify any patterns or correlations.

Bar chart:

A bar chart is a graphical representation of the frequency or proportion of categorical data. It shows how the values of a categorical variable are distributed across different categories. Bar charts are useful for comparing different categories and identifying patterns and trends.

Example Python code using Matplotlib to create a bar chart:

import matplotlib.pyplot as plt

import pandas as pd

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Create a bar chart of the frequency of the 'gender' column

plt.bar(data['gender'].value\_counts().index, data['gender'].value\_counts())

plt.xlabel('Gender')

plt.ylabel('Frequency')

plt.show()

Inference: The bar chart can help us understand the gender distribution in the dataset and identify any patterns or trends.

Box plot:

A box plot is a graphical representation of the distribution of a dataset. It shows the median, quartiles, and outliers of the data. Box plots are useful for identifying the spread and skewness of the data, as well as outliers.

Example Python code using Seaborn to create a box plot:

import seaborn as sns

import pandas as pd

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Create a box plot of the 'age' column

sns.boxplot(x=data['age'])

plt.show()

Inference: The box plot can help us understand the spread and skewness of the age distribution in the dataset, as well as identify any outliers.

**Conclusion**:

Exploratory Data Analysis (EDA) and visualization are important steps in data analysis that help to gain insights into the data, identify patterns, and make decisions. In this experiment, we used different EDA techniques such as data cleaning, data preprocessing, and data visualization to understand the dataset better.

We started by cleaning the data by removing missing values, handling outliers, and noisy data. Then we used different preprocessing techniques such as normalization and standardization to prepare the data for analysis.

We then used different visualization techniques such as histograms, scatter plots, bar charts, box plots, and line charts to understand the data better. Each visualization technique helped us to understand different aspects of the dataset, such as the distribution of data, relationships between variables, and trends over time.

By analyzing the visualizations, we were able to identify patterns and correlations between different attributes in the dataset. For example, we found that age and income are positively correlated, and males have higher income than females on average.

In conclusion, EDA and visualization are powerful tools that can help us to gain insights into the data, identify patterns, and make data-driven decisions. It is crucial to perform EDA and visualization on the data before applying any machine learning algorithms to ensure the accuracy and effectiveness of the model.